2015-09

Formal specifications for an electrical power grid system stability and reliability

Galinski, Jonathan J.

Monterey, California: Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

FORMAL SPECIFICATIONS FOR AN ELECTRICAL POWER GRID SYSTEM STABILITY AND RELIABILITY

by

Jonathan J. Galinski

September 2015

Thesis Advisor: Doron Drusinsky
Second Reader: Man-Tak Shing

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2015	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE
FORMAL SPECIFICATIONS FOR AN ELECTRICAL POWER GRID SYSTEM STABILITY AND RELIABILITY

5. FUNDING NUMBERS
HDTRA139119

6. AUTHOR(S) Galinski, Jonathan J.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943–5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Threat Reduction Agency (DTRA)
8725 John J. Kingman Rd.
Fort Belvoir, VA 22060

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

This thesis provides natural language requirements and associated formal specifications for an electric power grid. These specifications are the first step in using bounded constraint solving to detect early bleak states in an electric power grid system.

We analyze several methods of software verification and validation including Theorem Proving, Model Checking, and Execution-based Model Checking before determining that Execution-based Model Checking is the most suitable for specifying properties of a power grid. The requirements and specifications are broken into four categories: undesirable events, downward trends, failure to recover, and undesirable fluctuations. All specifications are focused on system stability and reliability as indicated by system frequency and operating in a secure N-1 state. Specifications from three out of the four categories were tested to ensure they meet the spirit and letter of the natural language requirements while eliminating ambiguity inherent to natural languages. Finally, we show how a Hidden Markov Model can be used to perform run-time monitoring in the presence of hidden states, thereby enabling run-time monitoring of systems where monitored artifacts are not all perfectly visible.

14. SUBJECT TERMS	**15. NUMBER OF PAGES**
run-time monitoring, statechart assertions, formal specifications, electric power grid, hidden markov model	63
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540–01–280–5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

FORMAL SPECIFICATIONS FOR AN ELECTRICAL POWER GRID SYSTEM STABILITY AND RELIABILITY

Jonathan J. Galinski
Captain, United States Marine Corps
B.S., United States Naval Academy, 2009

Submitted in partial fulfillment of the
requirements for the degree of

MASTER SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2015

Author: Jonathan J. Galinski

Approved by: Dr. Doron Drusinsky
Thesis Advisor

Dr. Man-Tak Shing
Second Reader

Dr. Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis provides natural language requirements and associated formal specifications for an electric power grid. These specifications are the first step in using bounded constraint solving to detect early bleak states in an electric power grid system.

We analyze several methods of software verification and validation including Theorem Proving, Model Checking, and Execution-based Model Checking before determining that Execution-based Model Checking is the most suitable for specifying properties of a power grid. The requirements and specifications are broken into four categories: undesirable events, downward trends, failure to recover, and undesirable fluctuations. All specifications are focused on system stability and reliability as indicated by system frequency and operating in a secure N-1 state. Specifications from three out of the four categories were tested to ensure they meet the spirit and letter of the natural language requirements while eliminating ambiguity inherent to natural languages. Finally, we show how a Hidden Markov Model can be used to perform run-time monitoring in the presence of hidden states, thereby enabling run-time monitoring of systems where monitored artifacts are not all perfectly visible.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

EMC	Execution-based Model Checking
ERCOT	Electric Reliability Council of Texas
FS	Formal Specification
FV&V	Formal Validation and Verification
HMM	Hidden Markov Model
HOL4	High Order Logic
MC	Model Checking
ML	Machine Learning
NERC	North American Electric Reliability Corporation
NL	Natural Language
PRC	Physical Response Capability
RV	Run-time verification
TP	Theorem Proving
UFLS	Under-Frequency Load Shedding
UML	Unified Modeling Language

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

This research was funded by a grant from the U.S. Defense Threat Reduction Agency (DTRA).

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. RESEARCH QUESTIONS

Mission-critical systems need to be highly dependable systems. Research has shown that formal specifications and formal methods help improve the clarity and precision of requirements specifications, which is a necessary ingredient of any highly dependable system. This thesis is focused on the analysis and validation of formal mission-critical requirements of an electric power grid system.

We will concentrate on the development of formal specifications that allow runtime monitors to detect bleak states. A bleak state is a system state where no assertion has failed yet, and it is a state from which the system will inevitably violate one or more formal specifications [1]. Computational tree logic-based model checking techniques can detect the existence of a bleak state, but cannot detect whether the bleak state is early or late [1]. Early bleak states refer to bleak states that are several states or more removed from the formal specification violation. A SAT-solver based bounded constraint system, however, has the ability to detect and identify early bleak states. While this thesis does not conduct bounded constraint solving to detect bleak states in an electric power grid system, it provides formal specifications that can be used for such verification technique. We will analyze the power grid system requirements and express the critical runtime behavior using first-order logic.

First, we identify observable operational setup and runtime patterns essential to the proper functioning of an electric power grid network system. Next, we generate natural language requirements based on the patterns and requirements identified. Once the natural language is specified, the requirements are formalized as statechart assertions, and converted into first order logic assertions. Scenario-based testing validates whether statechart assertions capture the intent of the natural language requirements [2].

B. THE NEED FOR RUNTIME VERIFICATION IN MISSION-CRITICAL SYSTEMS

Formal methods are known to improve software reliability and quality [3], [4]. Despite this and the positive development of formal methods over the years, acceptance and wide-spread use in industry and mission-critical systems has failed to materialize. Part of the research done in [5] identifies the multi-phase process of software development as a major reason for this. Never the less, formal methods of validation and verification, specifically execution-based model checking, have the ability to make an electric power grid system and other mission-critical systems more stable and reliable.

The need for formal methods is identified in key findings, causes, and recommendations of power grid blackouts. The investigation of the blackout on September 8, 2011, in Arizona and Southern California conducted by the North American Electric Reliability Corporation (NERC) identified lack of planning and inadequate situational awareness as the two overarching causes of the blackout [6]. This thesis provides a methodology to addresses the issue of situational awareness by presenting formal specifications for execution-based model checking. Run-time monitoring provides stability and reliability by providing adequate real-time situational awareness of conditions, a quality lacking according the NERC's findings. This topic is addressed again in Chapter III.

C. REMAINING CHAPTERS

Chapter II addresses the background information on formal validation and verification techniques and pertinent topics essential to understanding the methodology used in this thesis. Chapter III takes an in-depth look at an electric power grid system to identify parameters to utilize during run-time monitoring. Chapter IV defines the formal specifications created to monitor the stability of an electric power grid system. Chapter V describes the test suite used to validate the formal specifications provided in Chapter IV and discusses the results of running the formal specifications through the test suite. Section VI provides a proof of concept for utilizing a Hidden Markov Model to identify hidden data which can be used in behavioral and temporal pattern detection. Section VII

addresses future work regarding bounded constraint solving. Section VIII identifies shortcomings of this thesis and a conclusion.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. VISUAL TRADEOFF SPACE FOR FORMAL VERIFICATION AND VALIDATION TECHNIQUES

At the most basic level, each verification and validation technique needs to address two questions in regards to the reactive systems they are used to monitor. The first question is "what does the software need to do?" Answering this question requires knowledge of a system's functional requirements. Exploring the second question, "what must the software never do?" provides safe operational boundaries for a system. While traditional validation and verification techniques required a manual examination of requirements, modern program and system complexity have rendered manual examination insufficient and unfeasible. As a result we need to rely on automated validation and verification techniques to ensure system behaviors are correct.

In the scope of software engineering, verification refers to means taken to ensure a product is built correctly. Validation is the effort to guarantee the right product is built for a specific purpose [7]. To address the implementation and use of formal validation and verification techniques to capture setup and runtime requirements of an electric power grid, we identified the technique best suited to the task. This was accomplished by using the visual tradeoff space in [5], which provides a framework and comparison of three prevalent formal validation and verification techniques. The three techniques analyzed are theorem proving, model checking, and execution-based model checking. The framework provided is called the formal validation and verification tradeoff cube as shown in Figures 1 and 2. The tradeoff cube is comprised of the associated cost and coverage of each formal validation and verification technique. Factors contributing to the cost and coverage of each technique are the ability to specify complex properties, the effort required to create specifications of complex properties, and input effort required for software implementation. Cost refers to the financial cost required to generate and validate correct specifications [5].

In this section, we examine three validation and verification techniques, explain the makeup of the tradeoff cube, compare techniques using the tradeoff cube, and identify

the best technique to use for an electric power grid system. Each technique is evaluated in regards to cost and coverage space in three separate areas: specification/validation, program/implementation, and verification.

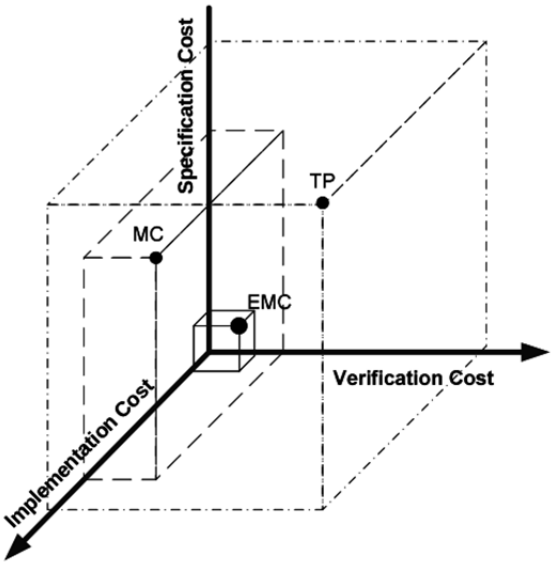


Figure 1. Cost Space

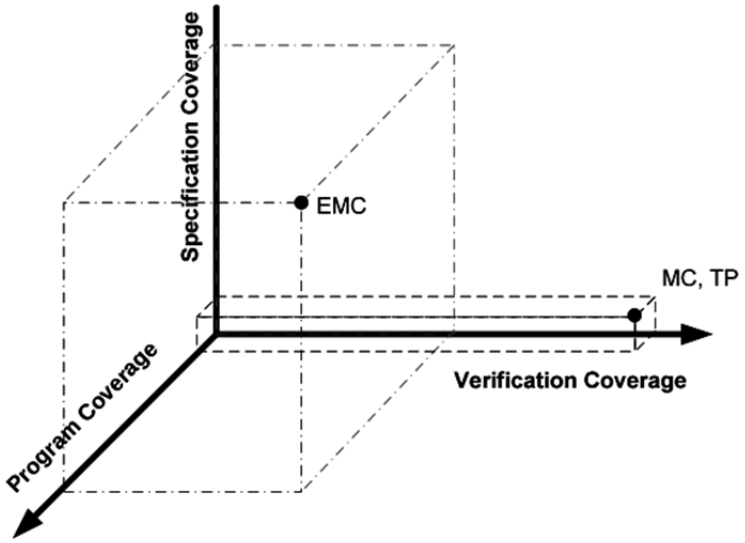


Figure 2. Coverage Space

1. Theorem Proving

According to [5], theorem proving is a formal verification technique that makes a convincing argument that a program meets a formal requirement through the use of mathematical proofs. One important aspect of theorem proving in regards to cost and coverage is that it requires a human driver. Additionally, the required level of expertise for the driver depends on the specification language employed. A driver monitoring a specification in Propositional Linear-time Temporal Logic requires a higher level of expertise than a driver observing a specification in Propositional-Logic. Existing methods employing theorem proving formal methods are A Computational Logic for Applicative Common Lisp, the Stanford Temporal Prover, HOL4 (High Order Logic), Prototype Verification System, and Type systems to name a few [5].

Theorem proving's specification dimension is dependent on the expressive power of the formal specification language chosen and subsequently how easy that language is to use. If a theorem prover is highly automated, it most likely has a restrictive, less expressive language. In general, theorem provers support relatively weak languages [5], cites this as the primary reason theorem proving techniques have low specification coverage with high specification cost.

Another downside of theorem proving is its reliance on special programming languages. When it comes to program implementation, the inability of these languages to interface with applications using prominent high-level languages like Java or C++ leads to low coverage with high cost. From a verification standpoint, the presence of a knowledgeable user assures suitable coverage. The required presence of a knowledgeable user, however, causes verification cost to be high in this context.

2. Model Checking

According to [8], model checking algorithmically analyzes a program to prove certain properties hold true. Once model checking is set-up for a program, no human driver with expertise in the appropriate specification language is required.

Model checking shares similar language limitations to theorem proving in the specification dimension. Both techniques have applications that use Propositional Linear-

time Temporal Logic as their specification language. Additionally, both techniques are text-based, making system visualization difficult for designers. Other applications of model checking rely on computational tree logic which uses path operators, complicating the formation of correct specifications. As a result, model checking is weak in the specification dimension. While model checking also has high cost in the specification dimension, it has a lower cost than theorem proving because it does not require a highly skilled driver to complete a proof process.

Model checking in the program/implementation dimension suffers from being limited to a finite-state component and the number of states in the component. This limitation results in the state-space explosion problem. According to [9], this phenomenon is when an increase in the number of processes leads to an exponential growth in the state space which model checking techniques conducting state enumeration cannot handle. The limited finite-state component comes from this. Additionally, the artifact used in the model checker is often not the same as the original system. Rather, it is a smaller segment or an abstraction of the overall system. This combined with the state-space explosion problem causes model checking to be weak and high cost in the program/implementation dimension.

Model checking promotes automatic verification on command without a driver. It does this while providing full verification coverage for anything within its finite-state component. Thus, model checking is strong and has low cost in the verification coverage dimension.

3. Execution-Based Model Checking

Execution-based model checking is divided into two separate parts: runtime verification and automatic test generation. Run-time verification refers to methods used to monitor a system or application and comparing its current behavior to formal specifications representing correct system behavior [5]. A high volume of automatically generated tests, used in conjunction with run-time verification of formal specifications, yields execution-based model checking.

While many runtime-verification tools utilize Propositional Linear-time Temporal Logic or Model Transformation Language as their specification language, many modern tools use more powerful, expressive, and easy to use languages such as StateRover's specification language and UML diagrams, which are the current state-of-practice. The availability and use of these languages allows execution-based model checking to be relatively powerful with low cost in the specification dimension.

Another advantage of run-time verification is that it is designed to be used with systems regardless of their size, complexity, or programing language used to create them. Because of this, execution-based model checking has high coverage and low cost in the program/application dimension. One weakness is that execution-based model checking relies on automatic test generation. This weakness materializes in the verification dimension. When the system under test and specification run concurrently, we cannot be assured the automatic test generator will generate a test that violates a requirement which means there cannot be full verification coverage. This causes execution-based model checking to be weaker in verification coverage than theorem proving or model checking. The more automated the automatic test generation tool is, the lower the cost in the verification dimension.

4. Best FV&V Technique for an Electrical Power Grid System

After weighing the three options available, execution-based model checking was selected as the most appropriate option for conducting formal validation and verification for an electric power grid for the following reasons:

- UML-statechart assertions allow for the visualization and easy implementation of mission and safety-critical requirements.
- Several specifications for electric power grids require monitoring time-series data. The specification dimension of coverage (Figure 2) indicates that both theorem proving and model checking using Propositional Linear-time Logic, their most expressive language, has a difficult time handling timing and time-series data. UML-statechart assertions, however, provide specification coverage in this area.
- An analysis of data monitored by a power grid requires high implementation coverage with low implementation cost. Execution-based model checking outperforms the other two techniques in this requirement.

- Despite having lower coverage in the verification dimension, the purpose of this thesis is to provide formal specifications for an electric power grid to be used to conduct bounded constraint solving to identify bleak states. The implementation of bounded constraint solving will improve coverage in the verification dimension while realizing the low cost and high coverage of execution-based model checking in other dimensions.

B. PATTERNS FOR TESTING FORMAL SPECIFICATIONS

An obstacle to converting from natural language to a formal specification is ambiguity. Natural language by definition has an element of vagueness making exact specification difficult and preventing computers from effectively analyzing it [10]. Formal specifications must capture the precise intent of the natural language requirement. [11] presents baseline patterns to consider when testing formal specifications. These patterns serve to validate a formal specification, making sure it does exactly what it means to do. A second benefit of these test scenarios is to ensure that the formal specification captures the intent of the natural language requirement.

The intent of a natural language requirement needs to be clear. To illustrate this, we convert one of our natural language requirements defined in Chapter IV into two different formal specifications. NL1 is the natural language requirement:

NL1. *Flag when frequency is less than or equal to 59.7 Hz for four minutes in a five minute period.*

R1 and R2 are different formal specifications that represent a legitimate interpretation of NL1:

R1: *Flag whenever more than N events E occur within one of a series of consecutive T intervals. N = 3, E= frequency <= 59.7, T = 5.*

R2: *Flag whenever more than N events E occur within one of a series of semi-consecutive intervals T. N = 3, E= frequency <= 59.7, T = 5.*

Figure 3 provides an example event time-line to evaluate R1 and R2. By observing the sequence it is clear that from event E at minute four through event E at minute seven, there are four instances of event E within a five-minute period. While R1 and R2 are legitimate interpretations of NL1, only R2 will flag at minute seven. The

counter in R1 resets after each five-minute interval, preventing it from identifying an instance of four consecutive E's, an instance NL1 seeks to identify. If an event that occurred every five minutes existed that would warrant resetting the counter, R1 would be the desired rule. However, the intent of NL1 is such that R2 does a better job of fulfilling it than R1 as demonstrated through an analysis of Figure 3.

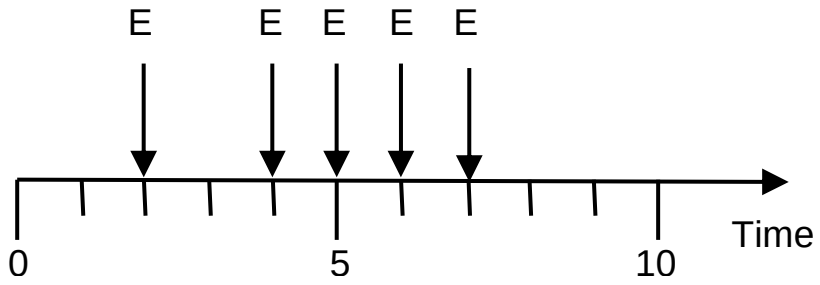


Figure 3. Event time-line for evaluating R1 and R2

THIS PAGE INTENTIONALLY LEFT BLANK

III. ELECTRIC POWER GRID

A. GETTING A GRIP ON THE POWER GRID

Before we discuss our formal specifications, we need a baseline understanding about power grids. More specifically, we need to understand the system's major components, how it operates, important values, metrics used to monitor it, and means for identifying and correcting undesired events.

The power grids of today are the largest engineered systems ever built. They are relied upon to deliver power on demand to the population of their respected areas. For the purpose of this thesis, we discuss the design and properties of the North American power grid which differs from other designs on several fronts to include evolution and selection of a single frequency. Alternating current is generally more desirable than direct current in regards to power grids because of inherit energy savings and because materials for alternating current are more conducive to transform between low voltage and high voltage [12]. Low voltage is used for consumption while high voltage enables long distance power transmission with low energy loss.

North America's bottom-up approach to power grid development focused on regionally strong grids connected by weaker links to nearby regions [12]. Our current U.S. transmission grid is a network of regional and local power authorities. The U.S. power grid as of 2009 consists of three independent regions. Each independent region is considered a large power grid network. These networks consist of two different types of networks: large-scale transmission grids and distribution grids. There is typically one transmission grid with numerous distributions grids, each covering a small area.

The transmission grid is a highly meshed network comprised of high-voltage (100 to 1,000kV) power lines. A standard transmission grid line is about 100 kilometers and the average node in a transmission grid has 2.5 line connections, which provides a level of resiliency to the system [12]. Centralized generating stations feed into the transmission grid producing 500 to 5000 MW of power. The transmission grid transports this power to substations that convert the power to a lower voltage for distribution to

customers. Substations mark the point where the transmission grid ends and distribution grids begin. Once the substation converts power from the transmission grid to approximately 10 to 30kV, distribution grids consisting of short, tree-like circuits carry the power the remainder of the way to customers.

Despite transporting voltage differing in orders of magnitude, transmission and distribution grids follow the same guidelines with respect to physics. The power carried across power lines is an oscillating electric current that is either real power or reactive power. The electric current produces real power when it is in phase with the oscillating voltage. Real power does useful work while reactive power, when current is ninety degrees out of phase with voltage, affects oscillating voltage throughout the network and does no useful work [12]. Despite this, electrical loads always use real power and in most circumstances use reactive power as well. Steady state of a system is achieved when power injection from generators, electric loads, and line loss are in balance. As loads change, power fluctuates between types, generators fail, or transmission lines fail, kinetic energy can be lost. When kinetic energy is lost, generators decelerate, which leads to a deviation in grid frequency. The more severe the loss in kinetic energy, the greater the deviation in grid frequency.

One of the most critical aspects about the North American power grid is its maintenance of a single frequency. The need for a single frequency is a result of the use of synchronized alternating current interconnections. While European nations chose 50 Hz, North Americans chose 60 Hz, which continues to be our standard [12].

1. Reliability

In a perfect world, transmission grids would maintain a constant 60 Hz with no deviations regardless of the load. In practice, however, equipment failure, transmission line resistance, fluctuations in power generated by renewable energy sources, and inherent delays in a generator's ability to adjust to changing power demands prevent a constant 60 Hz from being achieved [12]. Deviation from 60 Hz is inherent in the system and does not need to be zero but all efforts need to be taken to ensure the deviation is as small as possible. Smaller deviations are the result of line resistance and generators

adjusting to changing demand, while equipment failure and renewable energy source power fluctuations account for larger deviations. The primary gauge of reliability in a power grid system is the system frequency. System reserves are specifically saved to stabilize system frequency in the event of larger deviations. If system reserves are not used quickly enough or are not available when system frequency reaches a certain threshold, the undesirable response of Under-Frequency Load Shedding (UFLS) is triggered to aid the return to a steady state. For this reason, system frequency is a monitored value that must be evaluated against formal specifications for an accurate assessment of the reliability of an electric power grid.

2. Stability

The power grid assesses reliability and stability as often as every five minutes. In addition to monitoring system frequency, reliability and stability are measured against three standards: N-1 feasibility, transient stability, and voltage stability [11]. The system is considered to be in a feasible N-1 state when there is an achievable steady-state solution in the event that a generator or transmission line fails [13]. On hand reserves play the biggest role in identifying if a steady-state solution is achievable. If system reserves are not sufficient, the power grid cannot proactively initiate the return to a steady-state. As will be established in the evaluation of the September 8, 2011, blackout in Section B of this chapter, the minimum amount of reserves a system is required to maintain is equivalent to the amount of power needed to reach a steady state if the largest generator is taken out of the system. If reserves meet this requirement, the system is considered in a feasible N-1 state.

The measure of a system's ability to reach the N-1 feasibility steady state solution is known as transient stability [12]. Voltage stability ensures that the steady-state solution is sufficient to withstand changes in electrical loading. These criteria ensure that a steady-state solution exists, that it is attainable, and once reached it can be maintained if all else remains equal. As the power grid evolves and increases in complexity, these metrics may fall short of providing a stable and reliable system. Until they are deemed obsolete and more suitable metrics are created to deal with changes, these metrics continue to

serve the power grids of today and will for some time. This thesis uses formal specifications of system frequency and the N-1 feasibility contingency because they are essential to the proper operation of the power grid and are the easiest on which to maintain accurate measurements.

B. ARIZONA-SOUTHERN CALIFORNIA SEPTEMBER 8, 2011, BLACKOUT

Despite the existence of sophisticated controls, power grid blackouts occur regularly and in increasing numbers. While the number of blackouts for the North American power grid averaged seven per year until 1995, increasing system complexities and other factors caused this value to increase to 36 per year starting in 2006 [14]. This section focuses on the findings of one of the larger and more catastrophic instances of power grid failure in recent years. The findings identify the need for more robust monitoring and provide important metrics to monitor.

An 11-minute disturbance caused cascading outages in the Pacific Southwest and left around 2.7 million customers without power on September 8, 2011 [6]. The extent of the damage increases considerably when taking into account the traffic lights, schools, businesses, water and sewage pumping stations, and transportation effected by the outages. While not the sole cause of the outage, it was initiated by the loss of one 500 kV transmission line[6]. This line went down multiple times in the past without causing outages indicating the presence of other contributing conditions. When flows redistributed to account for the transmission line loss they caused voltage deviations and overloads on transformers, transmission lines, and generators leading to overall load shedding.

The first and foremost finding was that the system was not being operated in a secure N-1 state. This indicates a failure to meet the North American Electric Reliability Corporation's (NERC) mandatory reliability standards requiring the Bulk Electric System (BES) to remain in a reliable condition in the event a single contingency occurs. Loss of a generator, transformer, or transmission line is an example of such a contingency. Possessing the required reserves and functioning infrastructure to maintain stability in the

presence of a single contingency indicates being in a secure N-1 state [6]. Proper N-1 contingency planning ensures that a system can anticipate possible contingencies, adopt measures to maintain stability, and have available resources on hand to keep the system in equilibrium. The failure to operate in a secure N-1 state stems from inadequate operations planning and lack of real-time situational awareness. These are reoccurring causes in many power grid failures.

C. SOUTHWEST COLD WEATHER EVENT FEBRUARY 1–5, 2011

The cold weather event from February 1–5, 2011, caused 3.2 million Electric Reliability Council of Texas (ERCOT) customers to lose power. In contrast to root causes from the Arizona-Southern California blackout, ERCOT's system was operating in a secure N-1 state and under-frequency load shedding was conducted effectively, preventing a more catastrophic event from taking place. While internal problems were not the primary issue, a cold weather storm caused 193 ERCOT generating units to fail or operate at less than optimal levels over the course of the day on February 1, 2011 [15]. The 193 generation units accounted for a total loss of 29,729 MW out of an estimated daily load capacity of 52,673 MW. The loss in generators overwhelmed reserves, forcing ERCOT to execute 4,000 MW of controlled load shedding [15]. While any system can be improved and is susceptible in some degree to outside threats, the planning and contingency values adhered to by ERCOT minimized the damage caused by the arctic cold front during this week. Thus, many formal specifications presented in this thesis correspond to parameters and values identified by ERCOT for the stable running of their power grid. Effective run-time verification tools such as formal specifications provide a greater chance for losses to be mitigated in the future.

1. N-1 and N-2 Contingency Values

An important aspect of ensuring an effective N-1 state is to maintain the proper reserves. Responsive reserves are referred to as the Physical Response Capability (PRC). NERC's Reliability Standard BAL-002–0 R3 requires the balancing authority, in this instance ERCOT, to maintain a PRC to cover the loss of the largest contingency in the system [15]. The purpose of the PRC is to provide the system with responsive means of

restoring system frequency to 60 Hz in the event of abnormal frequency deviation. NERC minimum PRC level for safe operation of the system is the N-1 contingency reserve level. ERCOT's N-1 contingency reserve level was set at 1354 MW, the rating of their nuclear-powered generating unit [15].

As an added layer of protection and to account for ERCOT not being synchronously linked with other interconnections, they maintain a larger reserve than is required. The larger reserve calculated by ERCOT as their N-2 contingency is 2300 MW. This means that 2300 MW is the PRC required to prevent load shedding to maintain system frequency at 60 Hz in the event that ERCOT's system simultaneously loses its two largest generation sources [15]. Ideally, PRC will never fall below 2300 MW. Actual PRC typically surpasses 2300 MW. In fact, going into the first set of outages, PRC was 3100 MW. Despite being well above required reserves, 3100 MW was not sufficient to account for the 29,729 MW loss in generation capacity caused by weather.

2. Frequency and Load Shedding Values

While system frequency and PRC are separately monitored values, they are inherently tied together. System frequency maintains itself around 60 Hz. When frequency falls to 59.7 Hz or lower, however, it is considered a large deviation and reserves must be used to contain and restore the system back to 60 Hz. This is imperative because if system frequency reaches 59.3 Hz or lower, the first block of automatic under-frequency load shedding is automatically triggered [15]. The first block will conduct a controlled dump of five percent of the total load on the system. If the PRC is at inadequate levels, the system loses its responsive capability to prevent load shedding at 59.3 Hz. The 4000 MW of load shedding during the cold weather event occurred specifically because of this reason.

IV. FORMAL SPECIFICATIONS

Using the information gathered from Chapter III, we can generate our formal specifications. We define each specification first as a natural language requirement then convert the natural language requirement into a UML-statechart assertion using generic assertions provided at [16]. Our specifications are broken into four separate categories: undesirable events, downward trends, failure to recover, and undesirable fluctuations. While the specifications cover several vital aspects of a properly functioning power grid, they are not a comprehensive list of every aspect of the power grid. Instead, they show essential properties of an electric power grid can be expressed and evaluated using the state of practice for run-time verification in software engineering. For the purposes of clarity, we used the PRC contingency values and frequency guidelines as provided by ERCOT in [15]. Every system has equivalent values which can be used to make the provided specifications apply.

A. NATURAL LANGUAGE REQUIREMENTS FOR AN ELECTRIC POWER GRID

Natural language requirements lay the groundwork for conversion to less ambiguous assertions. The following specifications are broken into four categories: undesirable events, downward trends, failure to recover, and undesirable fluctuations.

1. Undesirable Events

Undesirable events are the most simplistic rules that focus on one instance where values reach levels they should not be at. If one of these rules is flagged it does not necessarily mean that other specifications will flag. However, if specifications in other categories flag, there is a high probability one or several of natural language requirements in this category have been flagged. These are the base rules compound specifications are built from. Rules 5–8 in this section are focused around the relationship between PRC and frequency. Reserves measured by the PRC are used to restore system frequency to 60 Hz when it falls to or below 59.7 Hz. It is essential for operators to know when the

PRC is below N-2 criterion and frequency reaches a point when reserves are required, thus reserves are required but not available.

The primary purpose of creating natural language requirements and formal specifications of a power grid is to use them in bounded constraint solving to detect bleak states. However, this is not the sole purpose of creating them. In addition to conducting formal verification, run-time monitoring of formal specifications is used for informational purposes. In several cases of presented requirements, one requirement is a stronger version of another. For example, natural language requirement 4 is a stronger version of natural language requirement 3. Thus, if natural language requirement 4 flags, we also know that natural language requirement 3 has been flagged.

If our sole intent was to create requirements to detect bleak states, we would only need natural language requirement 4. Natural language requirement 3, however, serves as an early warning indicator for natural language requirement 4. The system and its operators need to be aware when system frequency hits 59.7 Hz so steps can be taken to prevent frequency from falling to 59.3 Hz where under-frequency load shedding is triggered. Without natural language requirement 3, early warning and preventive steps to stabilize system frequency will fail to occur responsively. Additionally, in most cases when natural language requirement 3 flags, natural language requirement 4 does not.

1. Flag when PRC is less than 2300 MW (N-2 criterion).
2. Flag when PRC is less than 1354 MW (N-1 criterion).
3. Flag when system frequency falls to 59.7 Hz or lower.
4. Flag when system frequency falls to 59.3 Hz or lower.
5. Flag when PRC is less than 2300 MW (N-2 criterion) and system frequency is 59.7 Hz or lower.
6. Flag when PRC is less than 2300 MW (N-2 criterion) and system frequency is 59.3 Hz or lower.
7. Flag when PRC is less than 1354 MW (N-1 criterion) and system frequency is 59.7 Hz or lower.
8. Flag when PRC is less than 1354 MW (N-1 criterion) and system frequency is 59.3 Hz or lower.

2. Downward Trends

The downward trend category identifies the violation of a less severe threshold followed by the violation of a more extreme threshold. This event shows the undesirable downward trend of a particular value. Note that the more severe frequency threshold below is set to 59.5 Hz instead of 59.3 Hz. This is because once frequency 59.3 is met, automatic load shedding is conducted. The reason for identifying the downward trend is to identify it and prevent it from continuing to 59.3 Hz. That is why the second threshold used is 59.5 Hz.

9. Flag when PRC falls below 2300 MW (N-2 criterion) and subsequently falls below 1354 MW (N-1 criterion) within 30 minutes.

10. Flag when PRC falls below 2300 MW (N-2 criterion) and subsequently falls below 1354 MW (N-1 criterion) within 15 minutes.

11. Flag when system frequency falls to 59.7 Hz or below and subsequently falls to 59.5 Hz or below within 30 minutes.

12. Flag when system frequency falls to 59.7 Hz or below and subsequently falls to 59.5 Hz or below within 15 minutes.

3. Failure to Recover

Findings from the blackout identified one of their shortcomings as a failure to restore the system to a secure N-1 state. NERC continued to write that a secure N-1 state must be restored as quickly as possible but should not take longer to achieve than 30 minutes [6]. This category addresses the 30 minute requirement for both the N-1 and N-2 criteria monitored by ERCOT while creating an intermediate requirement of 15 minutes to provide early warning. Additionally, specifications addressing a system's failure to restore frequency are also included.

13. Flag when PRC falls below 2300 MW (N-2 criterion) and is not restored to 2300 MW or greater in 30 minutes.

14. Flag when PRC falls below 2300 MW (N-2 criterion) and is not restored to 2300 MW or greater in 15 minutes.

15. Flag when PRC falls below 1354 MW (N-1 criterion) and is not restored to 1354 MW or greater in 30 minutes.

16. Flag when PRC falls below 1354 MW (N-1 criterion) and is not restored to 1354 MW or greater in 15 minutes.

17. Flag when frequency reaches 59.7 Hz or below and is not restored to greater than 59.7 in 15 minutes.

4. Undesirable Fluctuations

Findings in both the September and February blackouts indicate that falling in and out of stable N-1, N-2, and frequency states indicated the system was having difficulty reaching lasting equilibrium. This category is designed to identify a potential situation when this is the case.

18. Flag when PRC is less than 2300 MW for four minutes (or more) in a 5 minute period.

19. Flag when PRC is less than 2300 MW for eight minutes (or more) in a 15 minute period.

20. Flag when PRC is less than 1354 MW for four minutes (or more) in a 5 minute period.

21. Flag when PRC is less than 1354 MW for eight minutes (or more) in a 15 minute period.

22. Flag when frequency is less than or equal to 59.7 Hz for four minutes in a 5 minute period.

23. Flag when frequency is less than or equal to 59.7 Hz for eight minutes in a 15 minute period.

24. Flag when frequency is less than or equal to 59.3 Hz for four minutes in a 5 minute period.

25. Flag when frequency is less than or equal to 59.3 Hz for eight minutes in a 15 minute period.

B. FORMAL SPECIFICATIONS FOR AN ELECTRIC POWER GRID

The following formal specifications are the UML-statechart interpretations of the natural language requirements in the previous section.

1. Undesirable Events

To convert the natural language requirements for this category, we used Rule 1 from [16].

Rule 1: *Flag whenever event P happens*. Figure 4 denotes the UML-statechart for Rule 1.

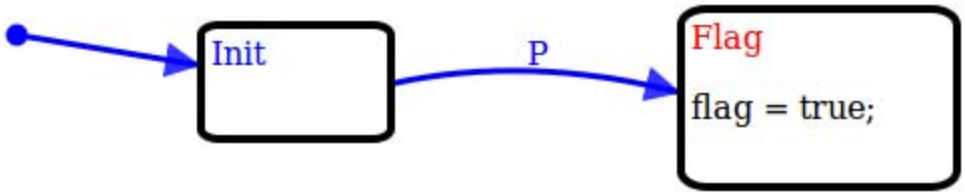


Figure 4. Rule 1 UML-statechart (from [16]).

We conform Rule 1 to our natural language requirements by creating formal specifications with the following assignments:

1. P = PRC<2300 MW
2. P = PRC<1354 MW
3. P = frequency<=59.7 Hz
4. P = frequency<=59.3 Hz
5. P = PRC<2300 MW ^ frequency<=59.7 Hz
6. P = PRC<2300 MW ^ frequency<=59.3 Hz
7. P = PRC<1354 MW ^ frequency<=59.7 Hz
8. P = PRC<1354 MW ^ frequency<=59.3 Hz

2. Downward Trends

To convert the natural language requirements for this category, we used Rule 11 from [16].

Rule 11: *Flag whenever event P with eventual event Q within time T after P.* Figure 5 denotes the UML-statechart for Rule 11.

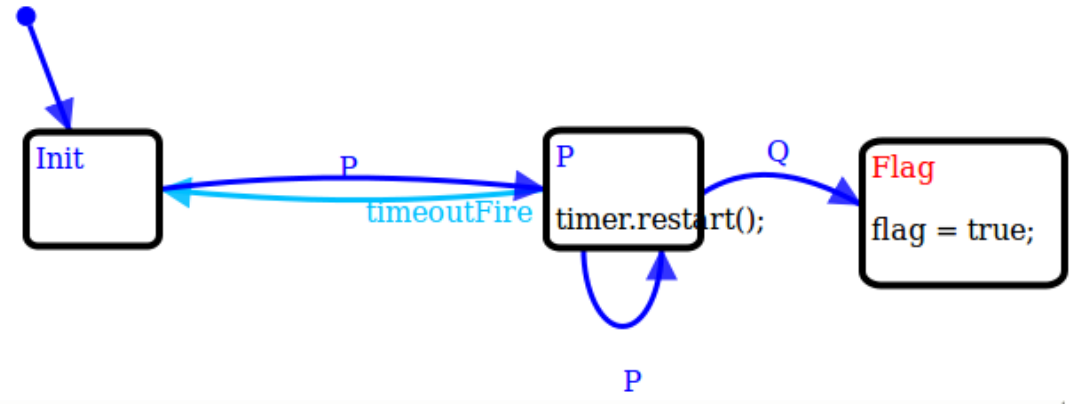


Figure 5. Rule 11 UML-statechart (from [16]).

We conform Rule 11 to our natural language requirements by creating formal specifications with the following assignments:

9. P = PRC<2300 MW, Q = PRC<1354 MW, T = 30 minutes
10. P = PRC<2300 MW, Q = PRC<1354 MW, T = 15 minutes
11. P = frequency<=59.7 Hz, Q = frequency<=59.5 Hz, T = 30 minutes
12. P = frequency<=59.7 Hz, Q = frequency<=59.5 Hz, T = 15 minutes

3. Failure to Recover

To convert the natural language requirements for this category, we used Rule 12 from [16].

Rule 12: *Flag whenever event P with no eventual event Q within time T after P*. Figure 6 denotes the UML-statechart for Rule 12.

We conform Rule 12 to our natural language requirements by creating formal specifications with the following assignments:

13. P = PRC<2300 MW, Q = PRC<=2300 MW, T = 30 minutes
14. P = PRC<2300 MW, Q = PRC<=2300 MW, T = 15 minutes
15. P = PRC<1354 MW, Q = PRC<=1354 MW, T = 30 minutes
16. P = PRC<1354 MW, Q = PRC<=1354 MW, T = 15 minutes
17. P = frequency<=59.7 Hz, Q = frequency<=59.7 Hz, T = 15 minutes

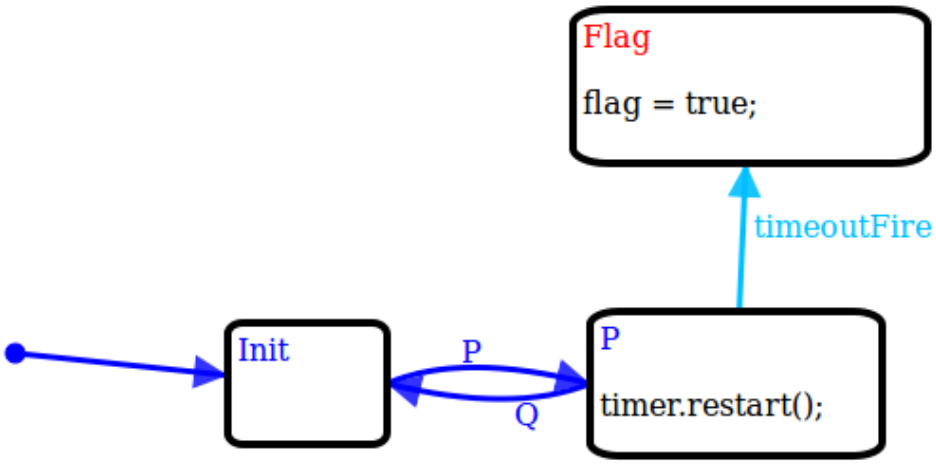


Figure 6. Rule 12 UML-statechart (from [16]).

4. Undesirable Fluctuations

To convert the natural language requirements for this category, we used Rule 28 from [16].

Rule 28: *Flag whenever more than N events E occur within one of a series of semi consecutive intervals T*. Figure 7 denotes the UML-statechart for Rule 28.

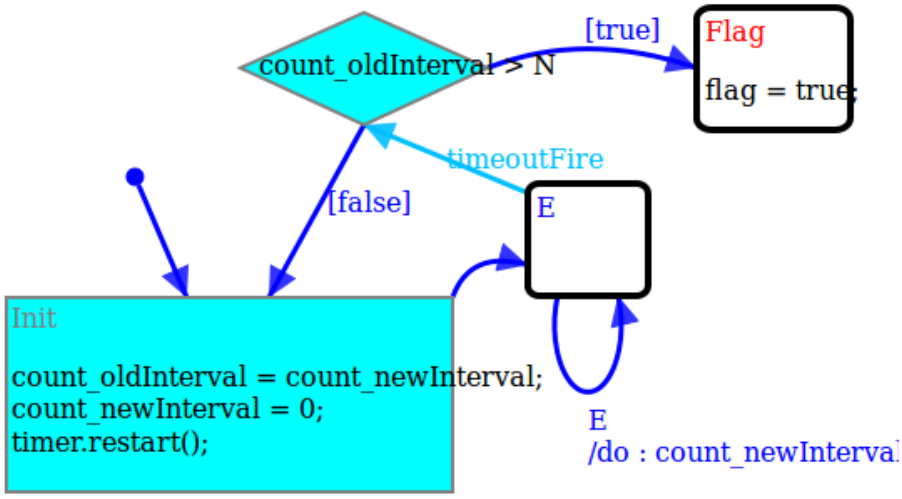


Figure 7. Rule 28 UML-statechart (from [16]).

We conform Rule 28 to our natural language requirements by creating formal specifications with the following assignments:

18. E = PRC<2300 MW, N = 3, T = 5 minutes
19. E = PRC<2300 MW, N = 7, T = 15 minutes
20. E = PRC<1354 MW, N = 3, T = 5 minutes
21. E = PRC<1354 MW, N = 7, T = 15 minutes
22. E = frequency<=59.7 Hz, N = 3, T = 5 minutes
23. E = frequency<=59.7 Hz, N = 7, T = 15 minutes
24. E = frequency<=59.3 Hz, N = 3, T = 5 minutes
25. E = frequency<=59.3 Hz, N = 7, T = 15 minutes

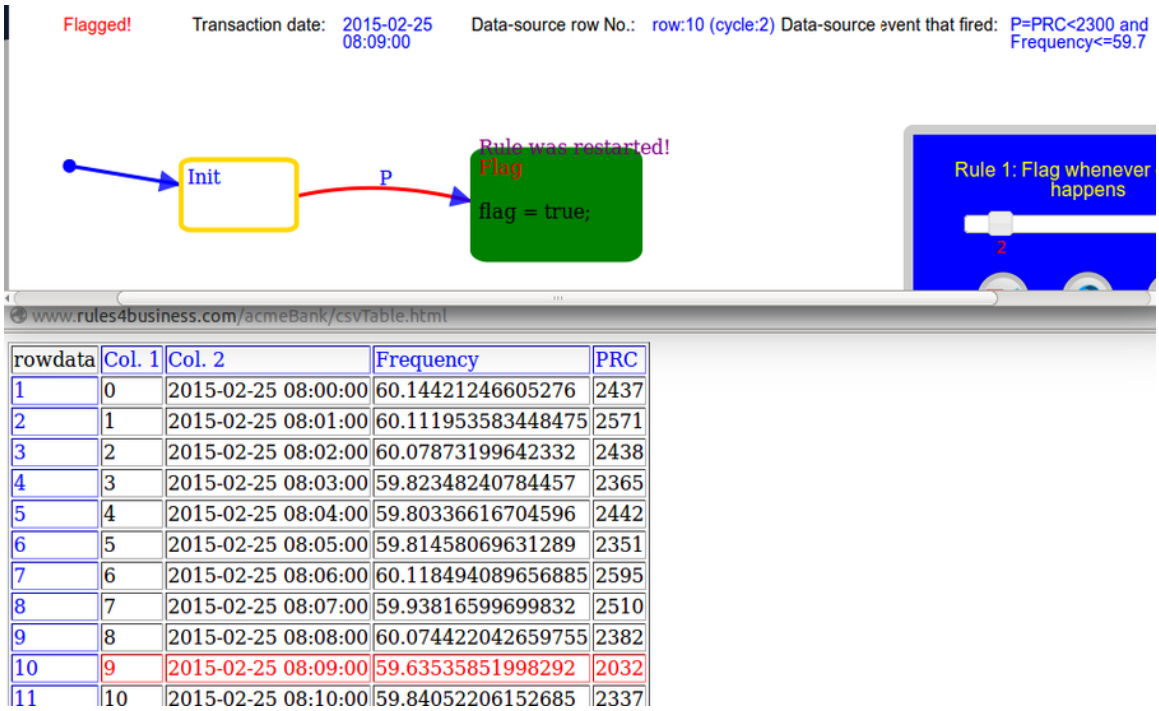
V. TESTING AND RESULTS

The next step to ensure our formal specifications are valid is to test them to see if they meet the spirit and letter of their associated natural language requirement. If the specifications achieve the spirit and letter then they are not only valid specifications but they have eliminated any ambiguity associated with the natural language requirement. We use [11] for guidance on test scenarios to accomplish this. We chose three compound specifications from different categories to execute test scenarios on. We conduct validation on specifications 5, 17, and 22.

A. SPECIFICATION 5

Recall that this specification identifies when PRC falls below 2300 MW and frequency is less than or equal to 59.7 Hz. To validate this specification we conducted five separate tests: obvious success, obvious failure, PRC is below 2300 MW but frequency is greater than 59.7 Hz, PRC is greater than or equal to 2300 MW but frequency is less than or equal to 59.7 Hz, and alternating between instances where PRC is below 2300 MW and frequency is less than or equal to 59.7 Hz. Time intervals do not come into play with this specification.

Obvious success is a scenario where we expect our specification to flag. Obvious failure is a scenario where we do not expect our specification to flag. To separate the obvious failure from subsequent tests we expect to fail, the data used for obvious failure does not contain a single instance where PRC is below 2300 MW or frequency is less than or equal to 59.7 Hz. Table 1 contains the pertinent data used for our obvious success test which is validated in execution down to the line number in Figure 8. The remaining four tests all failed to flag as expected.



rowdata	Col. 1	Col. 2	Frequency	PRC
1	0	2015-02-25 08:00:00	60.14421246605276	2437
2	1	2015-02-25 08:01:00	60.111953583448475	2571
3	2	2015-02-25 08:02:00	60.07873199642332	2438
4	3	2015-02-25 08:03:00	59.82348240784457	2365
5	4	2015-02-25 08:04:00	59.80336616704596	2442
6	5	2015-02-25 08:05:00	59.81458069631289	2351
7	6	2015-02-25 08:06:00	60.118494089656885	2595
8	7	2015-02-25 08:07:00	59.93816599699832	2510
9	8	2015-02-25 08:08:00	60.074422042659755	2382
10	9	2015-02-25 08:09:00	59.63535851998292	2032
11	10	2015-02-25 08:10:00	59.84052206152685	2337

Figure 8. Obvious success flag for specification 5.

Table 1. Obvious success data for specification 5.

Line No.	**Time**	**Frequency**	**PRC**
7	2015–02–25 08:7:00	59.938166	2510
8	2015–02–25 08:8:00	60.07442204	2382
9	2015–02–25 08:9:00	59.63535852	2032
10	2015–02–25 08:10:00	59.84052206	2337

B. SPECIFICATION 17

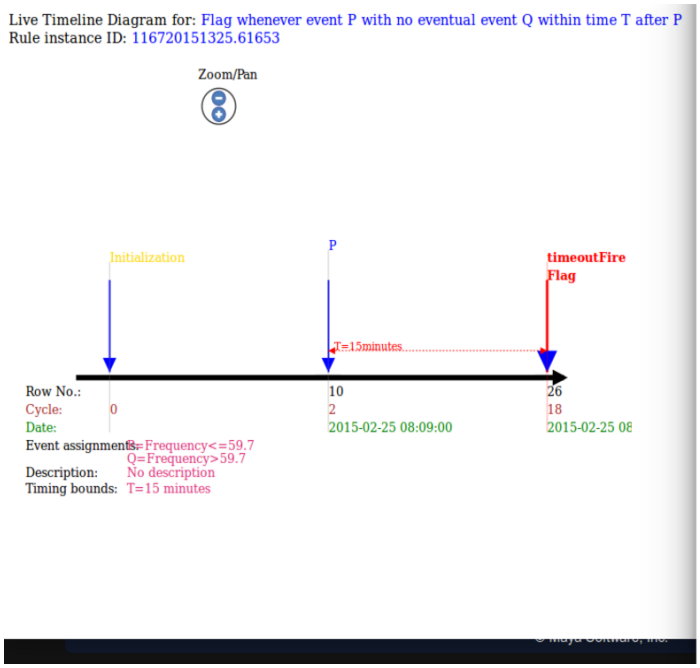
Remember that this specification evaluates whether system frequency recovers from frequency dropping to 59.7 or less within 15 minutes. Specification 17 aligns with several test scenarios provided in [11]. Thus, to validate specification 17 we conducted five separate tests: obvious success, obvious failure, event repetitions, and two sets of multiple time intervals. The one instance of multiple time intervals will flag after the first time interval while the other will flag after the second time interval. Success in validating this specification and eliminating ambiguity from its natural language requirement is

defined by a flag from the obvious success test and a single flag from both the multiple time interval tests.

Table 2 contains the pertinent data used for our obvious success test which is validated in Figure 9. Figure 10 shows the validation of the first multiple time interval test. The second multiple time interval test flagged as expected while the obvious failure and event repetition tests failed to flag as expected.

Table 2. Obvious success data for specification 17.

Line No.	Time	Frequency	PRC
8	2015–02–25 08:8:00	59.94136873	2550
9	2015–02–25 08:9:00	59.67921992	2490
10	2015–02–25 08:10:00	59.53028543	2438
11	2015–02–25 08:11:00	59.58969469	2420
12	2015–02–25 08:12:00	59.51353884	2345
13	2015–02–25 08:13:00	59.52348766	2513
14	2015–02–25 08:14:00	59.65914203	2418
15	2015–02–25 08:15:00	59.60677562	2449
16	2015–02–25 08:16:00	59.60178063	2406
17	2015–02–25 08:17:00	59.6808579	2303
18	2015–02–25 08:18:00	59.54540228	2540
19	2015–02–25 08:19:00	59.60511856	2416
20	2015–02–25 08:20:00	59.64173015	2490
21	2015–02–25 08:21:00	59.65411407	2435
22	2015–02–25 08:22:00	59.68877361	2479
23	2015–02–25 08:23:00	59.66904535	2558
24	2015–02–25 08:24:00	59.60590629	2477
25	2015–02–25 08:25:00	60.06379443	2450



rowdata	Col. 1	Col. 2	Frequency	PRC
1	0	2015-02-25 08:00:00	59.949605428310804	2471
2	1	2015-02-25 08:01:00	60.157994372909464	2335
3	2	2015-02-25 08:02:00	59.95310900006175	2441
4	3	2015-02-25 08:03:00	59.85708014578873	2387
5	4	2015-02-25 08:04:00	59.95701353075031	2438
6	5	2015-02-25 08:05:00	59.80299604426074	2347
7	6	2015-02-25 08:06:00	59.81577943551124	2538
8	7	2015-02-25 08:07:00	59.88019601856215	2303
9	8	2015-02-25 08:08:00	59.94136873047908	2550
10	9	2015-02-25 08:09:00	59.6792199222735	2490
11	10	2015-02-25 08:10:00	59.53028542875892	2438
12	11	2015-02-25 08:11:00	59.58969469387279	2420
13	12	2015-02-25 08:12:00	59.51353883778667	2345
14	13	2015-02-25 08:13:00	59.52348766322859	2513
15	14	2015-02-25 08:14:00	59.659142028736134	2418
16	15	2015-02-25 08:15:00	59.60677561587041	2449
17	16	2015-02-25 08:16:00	59.60178062952013	2406
18	17	2015-02-25 08:17:00	59.6808578997034	2303
19	18	2015-02-25 08:18:00	59.54540227788432	2540
20	19	2015-02-25 08:19:00	59.60511855853135	2416
21	20	2015-02-25 08:20:00	59.64173015235239	2490
22	21	2015-02-25 08:21:00	59.65411407354511	2435
23	22	2015-02-25 08:22:00	59.68877361390546	2479
24	23	2015-02-25 08:23:00	59.66904535407315	2558
25	24	2015-02-25 08:24:00	59.60590629461982	2477
26	25	2015-02-25 08:25:00	60.06379443335155	2450
27	26	2015-02-25 08:26:00	60.09617192872826	2413

Figure 9. Obvious success flag for specification 17.

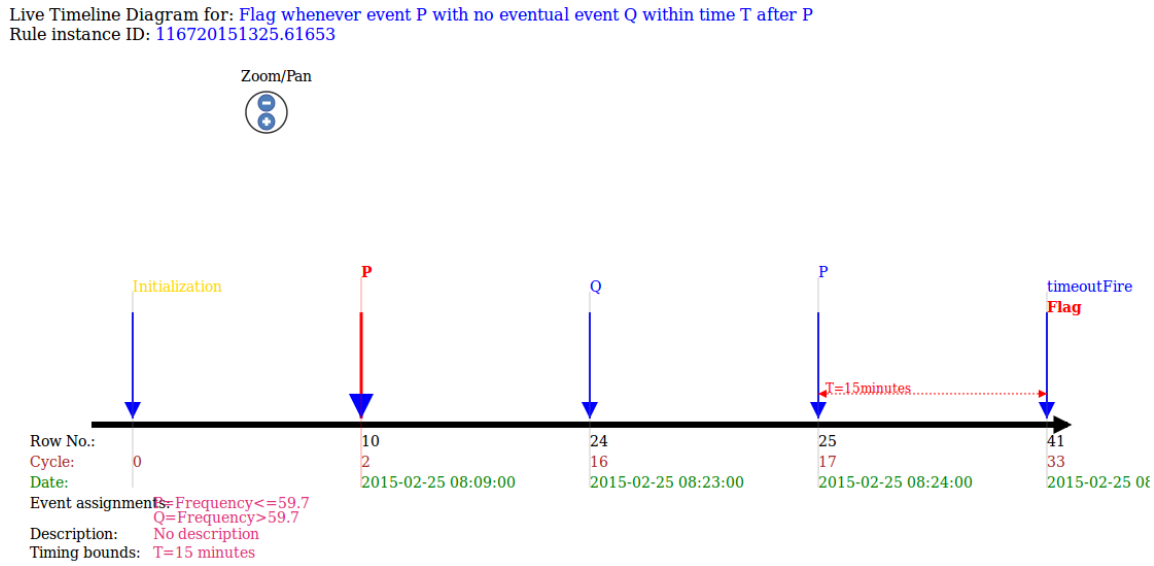


Figure 10. Multiple time intervals flag for specification 17.

C. SPECIFICATION 22

In Chapter II, we discussed the ambiguity associated with specification 22: Flag when frequency is less than or equal to 59.7 Hz for four minutes in a five-minute period. We also expressed that R2 did a better job that R1 of fulfilling the natural language

requirement. To validate that statement, we conducted the following five tests on the formal specification: obvious success, obvious failure, and three instances of multiple time intervals where we expect the first instance to flag once, the second instance to flag twice, and the third instance to flag three times.

Table 3 contains data equivalent to the timing shown in Figure 3 from Chapter II while Figure 11 shows run-time execution of the data and where it expectedly flags once. Obvious failure did not flag and the remaining instances of multiple time intervals flagged as expected, thus validating our assertion.

Table 3. Multiple time interval data for specification 22.

Line No.	Time	Frequency	PRC
1	2015–02–25 08:0:00	60.04426419	2346
2	2015–02–25 08:1:00	59.69104352	2344
3	2015–02–25 08:2:00	60.19712239	2461
4	2015–02–25 08:3:00	59.54800324	2434
5	2015–02–25 08:4:00	59.51057456	2343
6	2015–02–25 08:5:00	59.61708826	2367
7	2015–02–25 08:6:00	59.61302657	2495
8	2015–02–25 08:7:00	60.15421913	2440

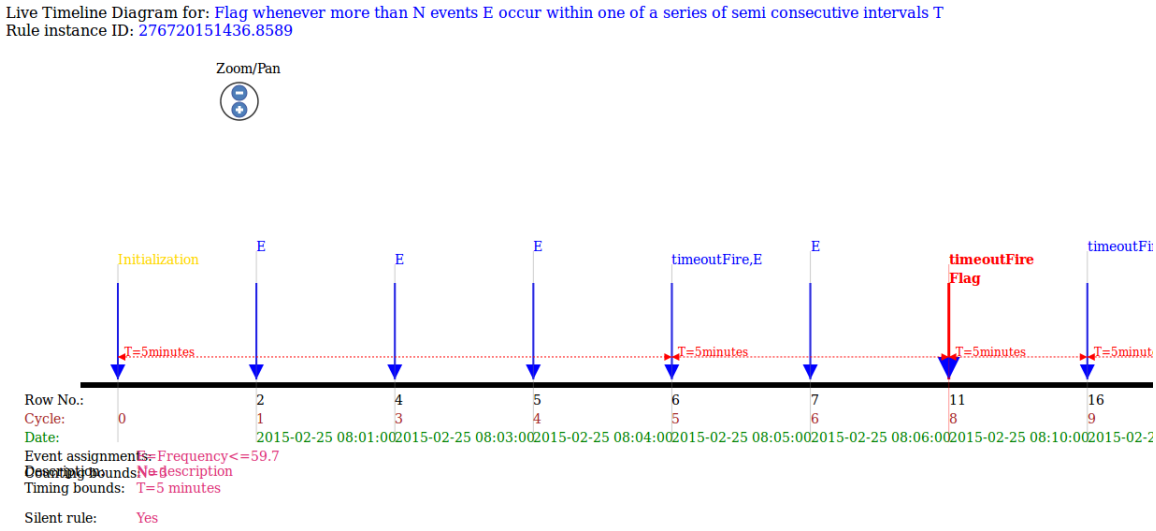


Figure 11. Multiple time intervals flag for specification 22.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. HIDDEN MARKOV MODEL

A. INTRODUCTION

This chapter provides a technique for the detection of behavioral and temporal patterns in data generated by an electric power grid with only partially necessary data in evidence.

At a minimum, an electric power grid is monitored by visible measurable data to include the date time, system frequency, and PRC. Fluctuations in system frequency and PRC are inherent within reasonable limits but larger changes are the result of any number of events and can indicate internal or external issues in the system. The nature of larger fluctuations, whether the loss of a large generator, the downing of a transmission line, or inherent variable energy flow from a renewable resource, is not explicitly available.

A well-known technique for behavioral pattern detection is the Hidden Markov Model (HMM) technique to learn and identify hidden artifacts. The HMM technique used in conjunction with probabilistic UML-based formal specifications at run-time provides us with a technique to conduct behavioral pattern detection and thus get data not inherently available. This system requires the end user to conduct a learning phase based off the system's deterministic patterns which is used to identify hidden artifacts in those patterns. Frequency and transition analysis of the identified artifacts provide the state set for the HMM. During run-time, monitored data is used by the HMM to identify hidden data. Once identified, the hidden data is used for probabilistic pattern detection and run against existing formal specifications.

HMM are state machines where transitions between states and a state outputs are probabilistic in nature. With this in mind, we generate a HMM by learning system states and their transitions [17]. Once created the HMM uses known data to identify a hidden system state and its sequences. This model does not use the probability of an observable sequence of states, rather it provides the probability of the system being in a particular state as the result of a series of observable data.

Run-time Verification refers to methods used to monitor a system or application and its behaviors by comparing current behavior to identified correct behavior specified by formal specifications [17]. The use of formal specifications ensures that during run-time, a system is operating within its intended bounds while providing a flag to identify deviation from normal and specify where the disturbance takes place to enable remedial actions.

We use the hybrid pattern detection technique proposed in [17]. The hybrid technique combines statistical pattern detection with formal specification and RV techniques. This technique is used to identify hidden states in an electric power grid and use the identification of these states to conduct RV based off an arbitrary formal specification. Section B of this chapter includes a formal specification for an electric power grid's hidden state, an explanation of the learning phase conducted to generate the HMM, the resulting HMM matrices and their implementation, and an explanation of run-time analysis of system state as generated by the HMM in regards to our presented specification.

B. MODEL GENERATION, IMPLEMENTATION, AND USE

This section discusses the hidden data states, a formal specification about one of these states, the learning process, the HMM, and how these factors fit together to enhance our ability to conduct RV of an electric power grid system. The goal of this system is to input a spreadsheet containing known data to a program executing the HMM, the program generating a "State" column to the spreadsheet, then running the spreadsheet against the provided formal specification to see if it flags. Unfortunately, running the HMM requires a code generator we do not have access to. Consequently, we generate the HMM parameters, explain the execution flow of conducting pattern matching for the transaction log and subsequently conducting RV with a formal specification, but we do not execute these actions.

1. Formal Specification for an Electric Power Grid's Hidden State

Known data such as date, time, system frequency, and PRC are integral values to monitor for an electric power grid. In earlier chapters, we generated formal specifications

that will facilitate RV of a system that is monitoring these values. As previously mentioned, small fluctuations in system frequency are inherent but larger fluctuations can be attributed to certain events that place the power grid in a specific state which is unobserved during run-time. Using changes in the system frequency and other knowledge of the power grid, we can identify hidden states that indicate healthy or detrimental run-time behavior. Four states are identified and used in this chapter: Steady State (S), Loss of a Generator (G), Transmission Line Down (T), and Variable Energy Flow from a Renewable Resource (R). Changes in system frequency can be directly tied to these states or events. If the change in system frequency is severe enough, system reserves represented by the PRC need to be used to meet a state of equilibrium. The state of this equilibrium is identified by S. Equilibrium takes into account minor, unavoidable fluctuations in system frequency.

As the worst case of our hidden states, power grid system state G, Loss of a Generator, will be the focus of our formal specification. The following is a sample natural language requirement about state G; if a situation conforms to this specification, it is considered to be flagged: *Flag when the system is in state G for more than 3 minutes in a semi-consecutive 5 minute interval.* To convert this NL specification to a UML statechart assertion, we utilized software provided at [16]. Rule 28 on the site conformed to our rule and was utilized to generate our assertion.

Rule 28: *Flag whenever more than N events E occur within one of a series of semi consecutive intervals T*. Figure 7 depicts the generic UML statechart for our rule. This rule is customized to our purposes by making the following assignments:

26. N = 3, E = State == G, T = 5 minutes.

Once we identify system state, we can conduct RV against this specification.

2. Learning Phase

In contrast to a machine learning (ML) approach, the learning phase in our method requires manual input from a user who can identify which of the four states the power grid system is in by line of data. One shortfall of this approach is that it is time consuming and does not generate the volume of data a ML approach can. Despite this, it

makes up for it with accuracy and being able to provide information a machine cannot. To generate our HMM, we used a spreadsheet accounting for a seven-day period. For a system recording data every minute on the minute, this provided 10,080 samples of hidden states and state transitions. Table 4 depicts a sample of learning phase data. The learning phase consisted of manually generating the "State" column of the spreadsheet. For the purposes of conducting a proof of concept, a small sample size was suitable. For a usable implementation, more data is required in this phase to achieve the most accurate HMM possible.

Table 4. Learning Phase.

Time	Frequency	PRC	State
08:05:00	59.680	2318	S
08:06:00	59.825	2401	S
08:07:00	60.074	2557	S
08:08:00	59.839	2474	R
08:09:00	59.431	2310	T

3. Hidden Markov Model

The learning phase determines the HMM parameters needed to conduct the pattern matching architecture for the transition log and formal specification. The HMM parameters are the set of states, an observable tuple describing potential data combinations, Matrix A, Matrix B, and initial state distribution [17]. The set of states are the four states previously mentioned: S, G, T, and R. The observable tuple, O, is a conjunction of frequency state and PRC state both represented as integers. Frequency is the first value in the tuple and PRC is the second. Table 5 shows frequency state integer assignments and Table 6 shows PRC state integer assignments. Thus, the tuple <4,5> represents the instance where frequency is between 59.9 to 60.1 Hz and PRC is between 2100 to 2300 MW. With seven states each, there are 49 possible observable tuples for each HMM state.

Table 5. Frequency State Assignment.

Frequency State	Frequency Range
0	Frequency <= 59.3
1	59.3 < Frequency <= 59.5
2	59.5 < Frequency <= 59.7
3	59.7 < Frequency <= 59.9
4	59.9 < Frequency <= 60.1
5	60.1 < Frequency <= 60.3
6	Frequency > 60.3

Table 6. PRC State Assignment.

PRC State	PRC Range
0	PRC < 1354
1	1354 <= PRC < 1500
2	1500 <= PRC < 1700
3	1700 <= PRC < 1900
4	1900 <= PRC < 2100
5	2100 <= PRC < 2300
6	PRC >= 2300

Using standard frequency analysis in conjunction with learning phase data, we generated Matrix A and Matrix B. Matrix A, represented as Table 7, provides the state transition properties for the HMM. Matrix B provides the probability of a given observable tuple O being observed in one of the four states: S, G, T, R. Table 8 represents part of Matrix B.

Finally, for the purposes of this chapter, we assume that the initial state is always S. With all HMM parameters accounted for, pattern-detection follows the process illustrated in Figure 12 [17]. Power grid data flows into the HMM where a probability estimation algorithm runs on the current iteration. The HMM outputs the vector of symbols and associated probabilities then becomes the input into the pattern's implementation code with the original power grid data. Pattern implementation code outputs a weighted version of a state-machine state change which can be used to conduct RV with existing formal specifications [17].

Table 7. Matrix A of HMM state transition probabilities.

Transition Source/Target	S	G	T	R
S	0.5533	0.0920	0.1540	0.2007
G	0.5831	0.3607	0.0023	0.0539
T	0.6263	0.0000	0.2981	0.0756
R	0.6580	0.0016	0.0332	0.3072

Table 8. A portion of Matrix B, the probability of observation O in a HMM state.

State	S	G	T	R
<1,0>	0.0082	0.0000	0.0185	0.0239
<1,1>	0.0037	0.0000	0.0057	0.0082
<1,2>	0.0072	0.0011	0.0085	0.0185
<1,3>	0.0131	0.0011	0.0178	0.0277
<1,4>	0.0196	0.0229	0.0264	0.0555

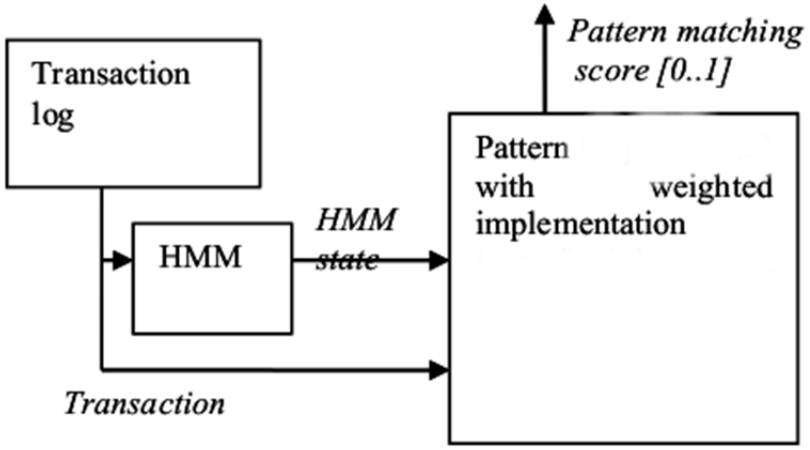


Figure 12. Pattern matching architecture for power grid data and requirement 26 (after [17]).

C. CONCLUSIONS ABOUT HIDDEN MARKOV MODEL

This chapter described how the methodology presented in [17] to find hidden information within financial data can be applied to hidden data with an electric power grid. While we were not able to run the HMM and conduct RV using hidden data, the

sampled application of the learning phase and generation of HMM parameters shows how the methodology can be applied to an electric power grid. There are areas that need to be developed before the presented model is ready to be used reliably in a system. The first area is state granularity. In addition to breaking up existing partitions into smaller segments, granularity can also be improved by adding more applicable data to the tuple. An example of this would be adding a Boolean value of 1 to represent peak hours and switching it to 0 during non-peak hours. This distinction has the potential to add new insight and accuracy into Matrix B, resulting in a more usable HMM.

The second area that needs to be developed moving forward is more lines of data generated in the learning phase. While a week's worth of data served our purposes, more data and higher quality data is required to ensure the model is accurate and precise. This chapter has shown that the hybrid pattern detection technique proposed in [17] for hidden financial data can be potentially applied to an electric power grid system.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSIONS AND FUTURE RESEARCH

We have demonstrated that formal specifications can be created to meet the spirit and letter of legitimate natural language requirements for a power grid. In doing so, the ambiguous nature of natural languages is negated. The purpose of this thesis is to lay the groundwork for detecting early bleak states by using bounded constraint solving. The intent behind identifying early bleak-states is to improve a mission-critical system's reliability and stability. Even without conducting bounded constraint solving, formal specifications provide a reliable tool for an operator's situational awareness. When used in conjunction, formal specifications and bounded constraint solving have the potential to significantly improve the dependability of mission-critical systems. The caveat is that research must be done to use the specifications from this thesis with the methodology from [1] to achieve the desired end-state of a more reliable system.

The adaptation of the HMM methods used in [17] provides another means for improving the electric power grid in the future. Research to improve the granularity and increase the amount of data used in the learning phase from Chapter VI also ties into detecting bleak states. If formal specifications on hidden states can be efficiently used to validate and verify the power grid system, they can be added to the group of formal specifications used to conduct bounded constraint solving.

Despite the accomplishments of this thesis, there are a few weakness and shortcomings. First and foremost, the specifications provided are not a comprehensive list of rules for the safe operation of an electric power grid. In reality, thousands of specifications, if not more, are required to achieve any level of assurance in such a complex system. Additionally, the 25 formal specifications in this thesis are focused on reliability and stability through several evaluations of system frequency and N-1 stability. While these are two important aspects, transient and voltage stability are also important criteria that were not included in the specifications.

The final shortcoming of this thesis is that real-world power grid data was not used at any time during the generation or testing of the formal specifications. While

manually generated scenario data was vital for specification validation and testing, going forward it would have been beneficial to interface with real world data. Additionally, access to ERCOT's data from the cold weather event in 2011 would have enabled us to evaluate the specifications responses to what happened and determine their practicality.

LIST OF REFERENCES

[1] D. Drusinsky, "Early detection of evolving system failures and temporal conflicts using parameterized formal specifications and bounded constraint-solving," *Innovations in Systems and Software Engineering*, vol. 11, no. 2, pp. 143–152, Jun. 2015.

[2] M. Alves, C. Bergue, D. Drusinsky, J. B. Michael, and M.-T. Shing, "Formal validation and verification of space flight software using statechart-assertions and runtime execution monitoring," *SoSE, 2011 6th International Conference on*, pp. 155–160, Jun. 2011.

[3] E. M. Clarke and J. M. Wing, "Formal methods: State of the art and future directions," *ACM CSUR*, vol. 28, no. 4, pp. 626–643, Dec. 1996.

[4] S. Easterbrook et al, "Experiences using lightweight formal methods for requirements modeling," *Software Engineering, IEEE Transactions on*, vol. 24, no. 1, pp. 4–14, Jan. 1998.

[5] D. Drusinsky, J. B. Michael, and M.-T. Shing, "A visual tradeoff space for formal verification and validation techniques," *Systems Journal, IEEE*, vol. 2, no. 4, pp. 513–519, Dec. 2008.

[6] Federal Energy Regulatory Commission and the North American Electric Reliability Corporation, *Arizona-Southern California Outages on September 8, 2011: Causes and Recommendations*, April 2012, http://www.ferc.gov/legal/staff-reports/04–27–2012-ferc-nerc-report.pdf (accessed November 11, 2014).

[7] P. Bourque et al., *The Guide to the Software Engineering Body of Knowledge*, 1st ed. Piscataway, NJ: IEEE Press, 1999, pp. 35–44.

[8] R. Jhala and R. Majumdar, "Software model checking," *ACM CSUR*, vol. 41, no. 4, pp. 21–22, Oct. 2009.

[9] E. Clarke et al., "Progress on the state explosion problem in model checking," *Informatics*, vol. 2000, 2001, pp. 176–194, Mar. 2001.

[10] K. Shimizu, D. L. Dill, and A. J. Hu, "Monitor-based formal specification of PCI," *Formal Methods in Computer-Aided Design*, vol. 1954, pp. 372–390, Jun. 2000.

[11] D. Drusinsky, J. B. Michael, T. W. Otani, and M.-T. Shing, "Validating UML statechart-based assertions libraries for improved reliability and assurance," in *SSIRI'08. Second International Conference on*, Yokohama, Japan, 2008, pp. 47–51.

[12] S. Backhaus and M. Chertkov, “Getting a grip on the electrical grid,” *Physics Today*, vol. 66, no. 5, pp. 42–48, May 2013.

[13] P. Kundur, *Power system stability and control*, 7th ed. New York: McGraw-hill, 1994, pp. 123–124.

[14] A. Mittal et al., “Real time contingency analysis for power grids,” in *Euro-Par 2011 Parallel Processing. 17th International Conference*, Bordeaux, France, 2011, pp. 303–315.

[15] Federal Energy Regulatory Commission and the North American Electric Reliability Corporation. *Report on Outages and Curtailments During the Southwest Cold Weather Event of February 1–5, 2011*, August 2011, http://www.ferc.gov/legal/staff-reports/08–16–11-report.pdf (accessed November 11, 2014).

[16] D. Drusinsky. “Rules for Business.” Rules4Business. http://www.rules4business.com/acmeBank/index.html (accessed June 5, 2015).

[17] D. Drusinsky, “Behavioral and Temporal Pattern Detection within Financial Data with Hidden Information,” *J. UCS*, vol. 18, no. 14, pp. 1950–1966, Jul. 2012.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California